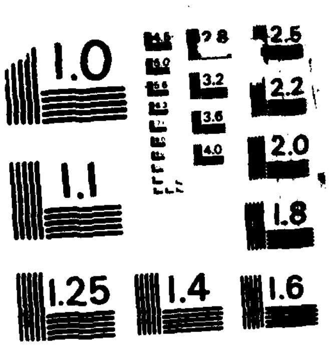
AD-A191 455   INTERACTIVE SYNTHESIS OF COMPUTER PROGRAMS(U) WEIZMANN INST OF SCIENCE REHOVOTH (ISRAEL) Z MANNA 31 MAR 87 DAJA45-84-C-0040   1/1

UNCLASSIFIED   F/G 12/5   NL

END
DATE
FILMED
5 88


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

DTIC FILE COPY (2)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188
Exp Date Jun 30, 1986

1a REPORT SECURITY CLASSIFICATION
Unclassified

1b RESTRICTIVE MARKINGS

AD-A191 455

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited

4 PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
R&D 4314-CC-01

6a. NAME OF PERFORMING ORGANIZATION
The Weizmann Institute of Science

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION
USARDSG-UK

6c. ADDRESS (City, State, and ZIP Code)
Rehovot 76100
Israel

7b. ADDRESS (City, State, and ZIP Code)
Box 65
FPO NY 09510-1500

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
USARDSG-UK

8b. OFFICE SYMBOL (If applicable)
ARO-E

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
DAJA45- 84-C-0040

8c. ADDRESS (City, State, and ZIP Code)
Box 65
FPO NY 09510-1500

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO
61102A	1L161102BH57	0	0

11. TITLE (Include Security Classification)
(U) Interactive Synthesis of Computer Programs

12. PERSONAL AUTHOR(S)
Professor Zohar Manna

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM 1.4.85 TO 31.3.87

14. DATE OF REPORT (Year, Month, Day)

15. PAGE COUNT
3

16. SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP
12	05	
17		

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Research concentrated on three topics:
1. Deductive synthesis of data flow networks,
2. Binary-search algorithms,
3. Theory of plans.
In the first area, a method of deductive synthesis of deterministic networks was developed. In the second area, general binary-search schema were constructed and specialized for particular applications. Finally, a variant of situational logic in which plans are explicit objects was introduced into the program - synthesis research.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☒ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a NAME OF RESPONSIBLE INDIVIDUAL
Dr. John M. Zavada

22b TELEPHONE (Include Area Code)
01-409 4423

22c. OFFICE SYMBOL
AMXSN-UK-RI

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

INTERACTIVE SYNTHESIS OF COMPUTER PROGRAMS

by

Zohar Manna
Professor of Computer Science
Weizmann Institute of Science
Rehovot 76100, Israel

Final Report:
United States Army Research
Contract DAJA-45-84-C-0040
April 1, 1985 - March 31, 1987

TECHNICAL SUMMARY

Our research concentrated on the following three topics. For each topic a copy of a published paper is enclosed.

- **Deductive Synthesis of Dataflow Networks**

  Jonsson, B., Z. Manna, and R. Waldinger, "Towards deductive synthesis of data-flow networks," *First Symposium on Logic of Computer Science*, Cambridge, MA, June 1986, pp. 26–37.

The synthesis of concurrent programs is much more complicated than the synthesis of sequential programs. In general, a concurrent program does not have a single input value and a single output value, but receives several inputs and sends several outputs during its execution. If we consider *sequences* of input and output values, then we can specify a concurrent program by giving a relation between the sequence of input values and the sequence of output values. This specification method is natural especially for networks of deterministic processes that communicate asynchronously by sending messages over buffered channels. Deterministic data flow networks fall into this category.

We have developed a method for the deductive synthesis of deterministic dataflow networks, which are specified by a relation between sequences of input values and sequences of output values.

Our synthesis method consists of two stages. The first stage, the deductive-synthesis stage, starts from a specification of the network. Using the deductive-tableau techniques of Manna and Waldinger, a system of recursive equations is synthesized. This system can be regarded as an applicative program that satisfies the specification for the network, but it does not directly represent any structure or parallelism of a network. In the second stage, the system of recursive equations is transformed into a dataflow network.

- **Binary-Search Algorithms**

  Manna, Z., and R. Waldinger, "The origin of the binary-search paradigm," *9th International Joint Conference on Artificial Intelligence*, Los Angeles, August 1985, pp. 222–224. Also in *Science of Computer Programming* Journal, Vol. 9, No. 1 (August 1987), pp. 37–83.

Some of the most efficient numerical algorithms rely on a *binary-search* strategy; according to this strategy, the interval in which the desired output is sought is divided roughly in half at each iteration. This technique is so useful that some authors (e.g., Dershowitz and Manna, Smith) have proposed that a general binary-search paradigm or schema be built into program synthesis systems and then specialized as required for particular applications.

It is certainly valuable to store such schemata if they are of general application and difficult to discover. This approach, however, leaves open the question of how schemata are discovered in the first place. We have found that the concept of binary search appears quite naturally and easily in the derivations of some numerical programs. The concept arises as the result of a single resolution step, between a goal and itself, using our deductive-synthesis techniques.

The programs we have produced in this way (e.g., real-number quotient and square root, integer quotient and square root, and array searching) are quite simple and reasonably efficient, but are bizarre in appearance and different from what we would have constructed by informal means. For example, we have developed by our synthesis techniques the following real-number square-root program $sqrt(r, \mathcal{E})$:

$$sqrt(r, \epsilon) \Leftarrow \begin{cases} \textit{if } max(r, 1) < \epsilon \\ \textit{then } 0 \\ \textit{else if } \left[sqrt(r, 2\epsilon) + \epsilon\right]^2 \le r \\ \qquad \textit{then } sqrt(r, 2\epsilon) + \epsilon \\ \qquad \textit{else } sqrt(r, 2\epsilon). \end{cases}$$

The program tests if the error tolerance ϵ is sufficiently large; if so, 0 is a close enough approximation. Otherwise, the program finds recursively an approximation within 2ϵ less than the exact square root of r. It then tries to refine this estimate, increasing it by ϵ if the exact square root is large enough and leaving it the same otherwise.

This program was surprising to us in that it doubles a number rather than halving it as the classical binary-search program does. Nevertheless, if the repeated occurrences of the recursive call $sqrt(r, 2\epsilon)$ are combined by common-subexpression elimination, this program is as efficient as the familiar one and somewhat simpler.

- **A Theory of Plans**

  Manna, Z., and R. Waldinger, "How to clear a block: A theory of plans," in *Reasoning About Actions and Plans*: *Proceedings of the 1986 Workshop*, Timberline, Oregon, July 1986, Morgan and Kaufmann (M.P. Georgeff and A.L. Lansky, eds.), pp. 11–45. Also in the *Journal of Automated Reasoning*, Vol. 3, No. 4 (December 1987), pp. 343–377.

Problems in commonsense and robot planning were approached by methods adapted from our program-synthesis research; planning is regarded as an application of automated deduction. To support this approach, we introduced a variant of situational logic, called *plan theory*, in which plans are explicit objects. A machine-oriented deductive-tableau inference system is adapted to plan theory. Equations and equivalences of the theory are built into a unification algorithm for the system. Frame axioms are built into the resolution rule.

Special attention was paid to the derivation of conditional and recursive plans. Inductive proofs of theorems for even the simplest planning problems, such as clearing a block, have been found to require challenging generalizations.

SCIENTIFIC COLLABORATION

The research was conducted in full collaboration with Dr. Richard Waldinger (Artificial Intelligence Center, SRI International), Eric Muller (programmer), and the following Ph.D. students of the Computer Science Department at Stanford University: Yoni Malachi (graduated 1986), Bengt Jonsson, Tomas Feder, Jon Traugott, Alex Bronstein, David Kashtan, and Marianne Baudinet.

Invited lectures and seminars, on the research reported here, were given in the following countries: Austarlia, China, England, India, Israel, Italy, Japan, Thailand, and in the USA.

DATE
FILMED
5 8